

Esquema de Plan Maestro para un Proyecto de Software

1. **Resumen Ejecutivo**

El proyecto consiste en desarrollar una aplicación web en Django que permita a los ciudadanos realizar solicitudes y comunicarse con el gobierno para expresar sus problemas

2. **Introducción**

El proyecto va dirigido para todas las personas que necesiten mandar alguna solicitud hacia el gobierno para que así puedan ser atendidas dichas solicitudes

3. **Objetivos del Proyecto**

Mejorar la transparencia gubernamental: Proporcionar un canal de comunicación directa entre los ciudadanos y el gobierno, aumentando la transparencia al permitir que las solicitudes y respuestas sean accesibles para todos.

Eficiencia en la gestión de solicitudes: Agilizar el proceso de gestión de solicitudes ciudadanas, reduciendo el tiempo y los recursos necesarios para manejarlas manualmente.

Participación ciudadana: Fomentar la participación activa de los ciudadanos en asuntos gubernamentales, brindándoles una plataforma para expresar sus preocupaciones y necesidades.

Reducción de la burocracia: Simplificar los procedimientos administrativos al digitalizar los trámites y la comunicación, lo que puede disminuir la burocracia y agilizar los procesos.

Mejora en la satisfacción del ciudadano: Ofrecer una forma conveniente y accesible para que los ciudadanos presenten sus solicitudes y obtengan respuestas, lo que podría aumentar la satisfacción general de los ciudadanos con los servicios gubernamentales.

Monitoreo y seguimiento: Facilitar el seguimiento y monitoreo de las solicitudes, permitiendo tanto a los ciudadanos como al gobierno hacer un seguimiento del estado y la resolución de cada solicitud.

4. **Alcance del Proyecto**

El software incluye una pagina principal en la que se puede ver una barra de navegación en la parte superior que contiene las siguientes opciones a elegir:

Crear solicitud: en esta pagina se encuentra una barra de navegación en la parte superior, y en la parte media se encuentra un cuadro que viene con un formulario con diferentes campos a rellenar, los cuales son: titulo, descripción, fecha de registro, estatus. Y al final de este cuadro se encuentra el botón de guardado.

Ver solicitudes: en esta pagina se muestra en la parte media un cuadro de texto que contiene las diferentes solicitudes que se han realizado por medio del formulario.

Iniciar sesión: al entrar a este enlace se te mostrara un cuadro en el cual tendrás que rellenar los campos de nombre de usuario y contraseña para poder acceder como servidor público.

5. **Arquitectura y Diseño**

Diagrama de casos de uso

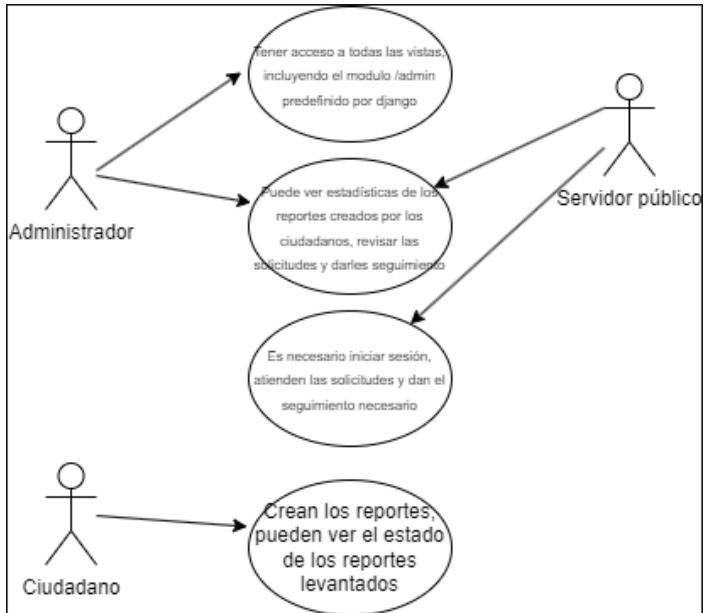
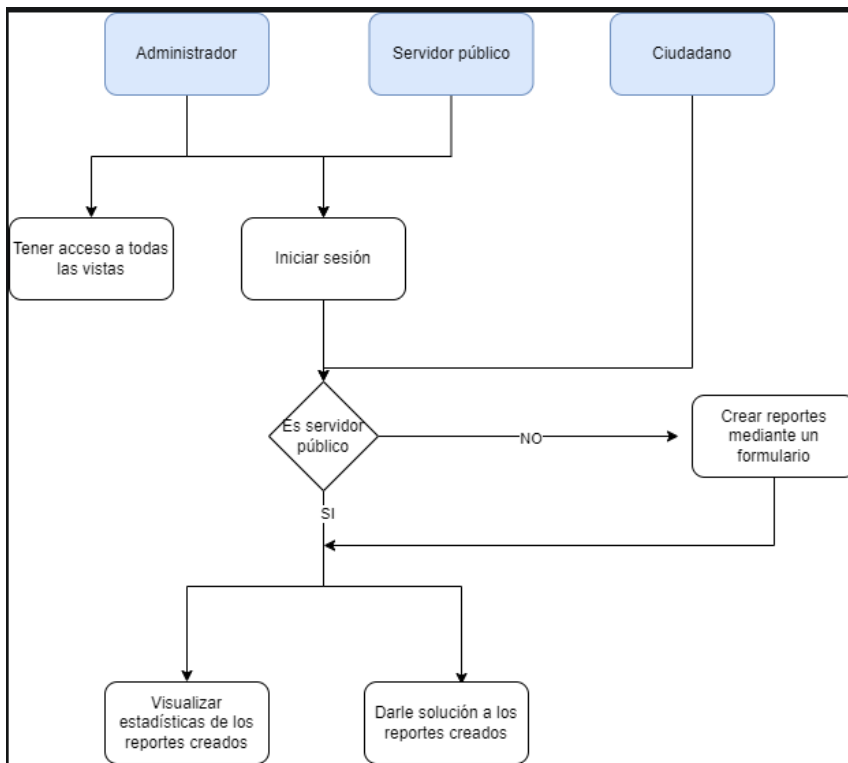


Diagrama de flujo:



Pagina de inicio de sesión

[Solicitudes](#)[Crear solicitud](#)[Ver solicitud](#)[Cerrar sesión](#)

Iniciar sesión

Usuario

Contraseña

Página principal(index)

[Solicitudes](#)[Crear solicitud](#)[Ver solicitud](#)[Cerrar sesión](#)

Crear solicitud

Titulo

Descripcion



Página de ver solicitudes

[Solicitudes](#)[Crear solicitud](#)[Ver solicitud](#)[Cerrar sesión](#)

Ver solicitudes

Solicitud 1

Solicitud 2

6. ****Desarrollo****

Metodología Cascada:

- Esta metodología sigue un enfoque secuencial en el que cada fase se completa antes de pasar a la siguiente.
- El proyecto comienza con la definición de requisitos, luego el diseño, la implementación, las pruebas y finalmente la entrega.
- Es adecuada cuando los requisitos son estables y bien entendidos desde el principio.

Metodología Ágil (por ejemplo, Scrum):

- El proyecto se divide en ciclos cortos llamados "sprints", generalmente de 2 a 4 semanas.
- Se comienza con una planificación del sprint, seguida de la implementación, las pruebas y la revisión.
- Permite la adaptación continua a medida que los requisitos evolucionan y se refina la solución.
- Se enfoca en la colaboración y la retroalimentación constante.

7. **Pruebas y Validación**

- Plan de pruebas detallado.
- Escenarios de prueba y casos de prueba.
- Criterios de aceptación.

El servidor y la base de datos fueron comprobados de que funcionan correctamente con el sistema de gestión de bases de datos MySQL, XAMPP, luego activando el ambiente virtual, seguido de esto, se ejecuta el comando `Python manage.py runserver` en la consola para así hacer funcionar la pagina web.

8. **Implementación y Despliegue**

Preparación del entorno de producción:

- Configurar el servidor y la infraestructura necesaria para alojar la página web en un entorno de producción seguro.
- Establecer una base de datos para almacenar las solicitudes y las respuestas del gobierno.

Compilación y pruebas finales:

- Asegurarse de que la aplicación esté lista para el despliegue realizando pruebas exhaustivas, incluyendo pruebas de funcionalidad, rendimiento y seguridad.
- Solucionar cualquier problema o error que surja durante las pruebas.

Empaquetado de la aplicación:

- Empaquetar la aplicación en un formato adecuado para su despliegue, que podría ser un archivo comprimido o una imagen de contenedor, dependiendo de la infraestructura utilizada.

Lanzamiento público:

- Anuncia el lanzamiento de la plataforma a los ciudadanos y ofrece instrucciones claras sobre cómo usarla.

9. **Mantenimiento y Soporte**

Realizar actualizaciones periódicas de la aplicación para agregar nuevas funciones, corregir errores y mantener la seguridad.

Escuchar los comentarios de los usuarios y realiza mejoras basadas en sus necesidades.

10. **Gestión de Riesgos**

Desarrollar estrategias para abordar cada riesgo identificado:

- Mitigación: Identificar acciones para reducir la probabilidad o el impacto del riesgo.
- Transferencia: Transferir la responsabilidad del riesgo a otro equipo o entidad, como mediante el uso de seguros.
- Aceptación: Reconocer el riesgo, pero no tomar medidas proactivas, generalmente aplicado a riesgos con impactos menores.
- Prevención: Tomar medidas para evitar que el riesgo ocurra en primer lugar.
- Contingencia: Preparar un plan de acción para manejar el riesgo si se materializa.

Implementar las estrategias de mitigación y otros planes diseñados para abordar los riesgos. Supervisar de cerca el progreso del proyecto y los riesgos identificados para asegurarse de que las estrategias estén funcionando.

11. **Presupuesto y Recursos**

1. Costos de Personal:

- Desarrolladores (2 personas durante 6 meses): \$X
- Diseñadores (1 persona durante 4 meses): \$Y
- Especialistas en Seguridad (1 persona durante 2 meses): \$Z
- Administradores de Sistemas (0.5 personas durante 6 meses): \$W

2. Costos de Infraestructura:

- Servidores y alojamiento: \$A
- Dominio y certificado SSL: \$B

3. Costos de Software y Herramientas:

- Licencias de software: \$C

4. Costos de Marketing:

- Publicidad y promoción: \$D

5. Contingencias:

- Margen del 10% para posibles gastos imprevistos: \$E